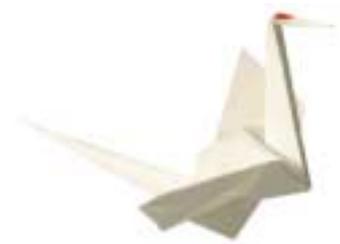


Information Management software

# IBM InfoSphere Streams

## 실시간 분석 처리의 재정의

**Roger Rea**  
IBM Software Group



---

## 목차

---

1. 서론
2. 스트림 컴퓨팅
3. 선별적 활용 사례
4. 구조 개요
5. Streams 2.0의 새로운 특징
6. 요약

## 총괄 요약

무어의 법칙, 즉 컴퓨터의 처리 능력이 2년마다 두 배씩 증가한다는 이 법칙이 두 가지 기술적 경향에 작용하고 있습니다. 서버에 사용되는 멀티코어 칩은 그 성능이 계속 향상되고 있으며, 센서 기기와 스마트폰, 태블릿의 임베디드 칩은 2년마다 데이터의 양을 두 배씩 늘리고 있습니다. 이 두 가지 기술 경향으로 인해 개발자들은 비즈니스 솔루션을 구현하기가 갈수록 더 어려워지고 있습니다. 일단 데이터의 양적 증가를 따라잡을 수 없으며, 멀티스레딩을 통해 멀티코어를 활용하기도 쉽지 않습니다(소위, “다수 코어의 문제”). 하지만 세계적 경쟁으로 인해 기업들은 더욱 신속한 운영과 의사결정을 추구할 수밖에 없습니다. 데이터의 양적 폭증과 의사결정 가능 기간의 단축에 직면하여, 기업들은 '진정한' 실시간 의사결정과 경쟁 우위 확보를 위해 분투하고 있습니다. LoB의 의사결정을 지원하는 기존 툴과 기술에서는 가장 먼저 데이터를 스토리지 장치에 기록하고, 그 후에 분석 쿼리를 실행하여 실질적인 통찰을 찾아내야 합니다. 영리한 기업들은 이러한 과정에서 발생하는 시간적 공백과 손실이 기회의 상실로 이어지며, 그것이 성공과 실패를 가를 수 있다는 사실을 빠르게 깨닫고 있습니다.

InfoSphere Streams는 데이터가 데이터베이스에 저장되기 전에 실시간 데이터 스트림으로부터 통찰적 정보를 찾아낼 수 있는 개발 툴과 실행 플랫폼을 제공함으로써 이러한 공백을 효과적으로 해소합니다.



## 서론

IBM InfoSphere Streams의 목적은 상당히 많고 다양하지만, 중요하지 않을 가능성이 있는 데이터로부터 적절한 데이터를 분리하여 정보와 지식을 적극적으로 분석 및 관리할 수 있는 혁신적 기능을 제공합니다. 구체적으로 말하면, InfoSphere Streams는 몇 가지 기술적 과제를 동시에 해결하여 최첨단 정보 기술의 가능성을 급격히 확대하고 있습니다.

- 사건 및 요건 변화에 실시간으로 대응합니다.
- 기존 시스템보다 수십 배 빠른 속도로 지속적으로 데이터를 분석합니다.
- 데이터의 형태 및 유형 변화에 신속히 적응합니다.
- 새로운 스트림 패러다임의 고가용성, 이질성 및 분산을 효과적으로 관리합니다.
- 공유된 정보에 보안성과 기밀성을 제공합니다.

일부 연구 기관, 오픈소스 진영 및 영리 기업을 가운데 이러한 기술적 과제 영역을 각기 분리해 해결하려는 시도가 있으나, 이 문제 영역들을 모두 동시에 해결하고자 시도하는 프로그램은 InfoSphere Streams 외에는 없습니다. InfoSphere Streams는 이러한 과제의 해결을 가로막는 여러 가지 근본적 장애물을 해소하고 있습니다. 미국 정부와 IBM 사이의 협력으로 2003년에 시작된 이 프로젝트는 여러 기관/단체를 통해 구현되어 공공 부문, 통신, 금융 시장, 에너지, e-Science, 의료 등 많은 부문에 다양한 애플리케이션을 구축했습니다.

## 스트림 컴퓨팅

스트림 컴퓨팅은 새로운 패러다임입니다. “전통적” 프로세싱 방식의 경우, 과거의 데이터에서 분석 쿼리를 실행했습니다. 예를 들어, 걷는 동안 GPS(Global Positioning System) 위치 데이터를 전송하는 가입자들의 데이터 세트로부터 지난 달 걸은 거리를 계산하는 방식입니다. 스트림 컴퓨팅에서는 GPS 데이터의 위치 정보가 갱신되는 매 순간마다 지속적으로 합산을 실행하는 “연속적 쿼리”와 유사한 프로세스를 실행할 수 있습니다. 첫 번째 경우, 과거 이력 데이터를 대상으로 질문이 이루어지며, 두 번째 경우 정적 질문에 의해 데이터가 지속적으로 계산됩니다. InfoSphere Streams는 이러한 연속적 분석을 시간에 따라 수정 가능하게 함으로써 한 걸음 더 전진하고 있습니다.

이 차이를 그림으로 간단히 살펴보면 다음과 같습니다.

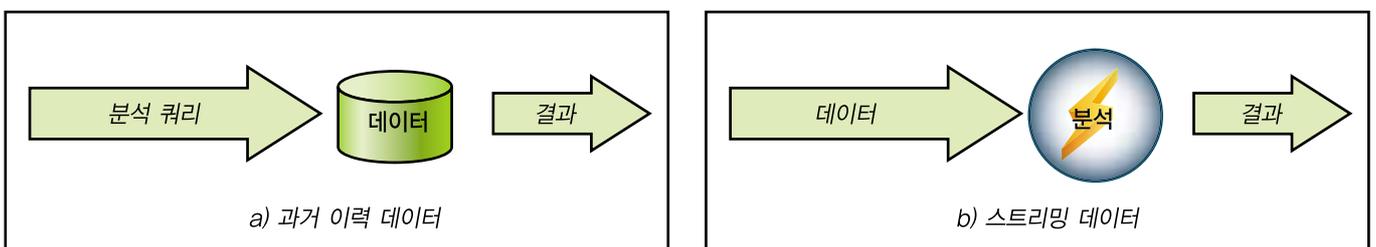


그림 1: 과거 이력 데이터와 스트리밍 데이터의 비교: 개념도



스트림 컴퓨팅 패러다임을 수용하는 다른 시스템들이 있지만, InfoSphere Streams는 연속적 프로세싱에 근본적으로 다른 차원의 접근 방식을 취하고 있으며, 분산형 런타임 플랫폼과 프로그래밍 모델, 연속적 분석 애플리케이션 개발 툴 등을 통해 차별화되고 있습니다. InfoSphere Streams는 센서, 카메라, 뉴스피드, 증권 시세, 전통적 데이터베이스 등 기타 다양한 소스에서 발생하는 데이터 스트림을 이용할 수 있습니다.

## 선별적 활용 사례

지난 몇 년 동안 InfoSphere Streams 용도로 수백 개의 애플리케이션이 개발되었습니다. 다음은 그 가운데 몇 가지 애플리케이션을 요약한 것으로서, InfoSphere Streams에서 지원하는 활용 형태를 잘 보여줍니다.

**통신:** 기술과 비즈니스 사이의 간극을 해소하는 과제는 아시아 지역 이동통신 사업자들에게 특히 두드러진 문제였습니다. 휴대폰에 임베디드 칩이 채택되면서 이메일, 텍스트, 사진, 동영상 송수신 및 Facebook 등의 소셜 사이트를 활용하는 정보 공유가 가능해졌습니다. 전화를 걸고, 메일을 보내고, 웹을 검색하거나, 텍스트 메시지를 이용할 때마다 휴대폰 스위치에서는 CDR(통화 상세 내역)을 내보냅니다. 데이터 손실을 방지하기 위해 스위치에서는 각 트랜잭션마다 두 개의 CDR을 방출하며, 따라서 나중에 중복을 제거하여 과금 처리 지원 시스템에 보내야 합니다. 그런데 데이터의 양적 증가로 인해 CDR을 적시에 수행하는 것이 갈수록 어려워지고 있습니다. 번호 이동으로 인해 가입자들은 언제든지 경쟁사로 옮겨갈 수 있으며, 일부 사용자들은 판촉 서비스를 이용하기 위해 하루에도 몇 번씩 통신사를 교체하기도 합니다. 따라서 통신 사업자들은 CDR 처리에 소요되는 시간을 거의 실시간에 가깝게 단축해야 할 뿐만 아니라, 어느 고객이 경쟁사로 이동할 것인지 예측하는 분석 작업도 동시에 수행해야 합니다. 고객의 행태에 대한 이와 같은 실시간 통찰을 통해 통신 사업자들은 고객 유지율을 높이는 조치를 취할 수 있는 것입니다.

InfoSphere Streams와 그 기민한 프로그래밍 모델로 고객사들은 지연 시간을 매우 낮은 수준으로 줄이면서 대량의 CDR을 처리하고, 이와 동시에 데이터를 분석할 수 있습니다. 어느 회사에서는 초당 최대 100,000건의 CDR이 처리되고 있으며, 단말간 지연 시간은 1초 미만입니다. 각 CDR이 수십억 개의 기존 CDR에 대조 확인되면서 실시간으로 중복이 제거되고 있으며, 이에 따라 데이터베이스에 저장되는 데이터의 양이 절반으로 줄었습니다. 이러한 사례는 Streams의 핵심적 용도인 실시간 동시 처리와 필터링 및 분석 기능을 잘 보여줍니다. 고가용성, 자동 내결함 기능 및 복구 기능이 실시간 상황 요약 기능과 함께 활용되어 IT 운영이 개선됩니다. 또한 실시간 CDR 분석이 곧 구현되어 비즈니스 운영이 더욱 개선될 것으로 예상됩니다.

**공공 기관:** InfoSphere Streams의 핵심적 장점은 데이터양이 많은 스트림을 분석하여 더 깊이 조사할 가치가 있는 소수의 항목을 파악할 수 있다는 점에 있습니다. 그 활용 사례 중 하나를 사이버 보안 분야에서 찾을 수 있습니다. 봇넷은 자체적으로 자동 실행되는 소프트웨어 에이전트 또는 로봇으로 구성된 네트워크를 말합니다. 봇넷은 C&C(통제 지휘) 컴퓨터에 응답하는데, 이러한 방식을 통해 범죄 용도로 봇넷에 대한 유료 액세스를 제공하는 지하 경제가 발생합니다. 이러한 봇넷은 급속히 진화하면서 네트워크 형태와 암호화 방식을 빠르게 바꾸기 때문에 탐지하기가 매우 어렵습니다. [Shadowserver Foundation](#)에서는 알려진 봇넷을 파악하여 C&C 컴퓨터는 수천 대, 봇 컴퓨터는 수만 대가 있을 것으로 추정하고 있습니다. InfoSphere Streams는 초당 100메가바이트가 넘는 IP 트래픽과 시간당 1,000만 개가 넘는 DNS 쿼리를 분석하여 급속히 변동하는 봇넷 경보를 생성하는 데 사용되었습니다. Streams는 기계 학습 모델뿐만 아니라, IBM InfoSphere Warehouse의 봇넷 경보 관련 이력 데이터를 이용해서 만들어진 모델도 활용합니다. 이러한 이력 모델의 생성에는 SPSS Modeler가 사용됩니다. Streams는 또한 모델 동향을 모니터링 하여 공격 패턴이 바뀌면 이력 데이터에 맞춰 모델 갱신을 요청하는 명령을 보내 탐지 모델을 최신 상태로 유지합니다. 이러한 활용 사례는 이력 데이터 및 실시간 데이터 분석을 위한 마이닝 모델의 지속적 업데이트를 잘 보여줍니다.



**금융 서비스:** 금융 서비스 산업의 여러 부문에서는 실시간에 가까운 사업 및 거래상의 결정을 위해 대량의 데이터를 신속히 분석하는 것이 매우 중요합니다. 현재 이러한 금융기관에서 일상적으로 처리하는 시장 데이터의 양은 메시지 기준 초당 백만 건 이상으로, 불과 1년 전 최대량의 두 배에 이릅니다. 시장 데이터의 이러한 급격한 증가는 예측 가능한 범위 내에서는 앞으로도 계속 이어질 것으로 예상되는데, 이는 현재 활용 가능한 여러 기술의 역량을 뛰어넘는 수준입니다. 앞서 가는 기업들은 자동 분석의 범위에 다른 유형의 데이터를 포함시켜 전략의 확대 및 정교화를 추진하고 있으며, 소스의 범위는 첨단 일기 예보 모델부터 방송 뉴스에 이르기까지 다양합니다. IBM에서 개발한 InfoSphere Streams 기반 트레이딩 프로토타입은 11개의 x86 블레이드 컴퓨터를 기반으로 확장형 트레이딩 애플리케이션을 호스팅 할 수 있으며, 기록된 최고 속도의 21배에 이르는 OPRA 데이터 피드를 처리할 수 있습니다.

**건강 모니터링(의료/보건):** 스트림 컴퓨팅을 활용하면 간호사와 의사의 업무 부담을 줄이면서 의료 분석 속도를 높일 수 있습니다. 개인 정보를 보호하면서 의료기관의 데이터 스트림을 분석하여 질병의 징후를 조기에 감지하고, 여러 환자 사이의 상관 관계와 치료 효과를 탐지할 수 있습니다. 이 분야의 경우, 시스템을 흐르는 데이터가 어떻게 도출되었는지를 추적하는 과정에서 데이터의 기원과 출처가 매우 중요합니다. IBM과 온타리오 대학 기술연구소 (University of Ontario Institute of Technology)가 수행한 이 분야 최초의 협력 작업에서는 InfoSphere Streams를 이용하여 신생아실의 조산아를 모니터링 했습니다. 데이터는 캐나다 토론토의 어느 병원에서 1년 반 동안 수집되었습니다. 또한 미국의 어느 병원에서는 동일한 분석 루틴을 이용하여 1년간 원격 측정이 이루어졌습니다. 그리고 올해 초에는 중국과 호주의 병원에서 구현 작업이 시작되었습니다.

**운송:** IBM은 아일랜드 더블린의 IBM Smarter Cities Technology Centre에서 InfoSphere Streams를 이용하여 버스로부터 1분마다 GPS 데이터를 수신하는 애플리케이션 작업을 진행 중입니다. 실시간 화면에는 이 도시에서 운행되는 모든 버스가 표시됩니다. 이 시범 프로젝트에서는 이를 확대하여 정거장마다 각 버스의 도착 시간에 관한 실시간 예측 정보를 제공하는 작업도 진행 중입니다. 이러한 정보는 버스 이용객들이 정거장 이용 계획을 세울 때 활용되어 기다리는 시간을 줄이는 데 큰 도움이 될 것입니다. 향후에는 개인용 교통 계획 프로그램을 통해 실시간 교통 정보에 기반한 권장 노선 정보를 제공하게 될 것입니다.

또한 환경 감시(산불 감시, 유량 관찰 등), 에너지 및 공공설비 산업(스마트 그리드의 싱크로페이저 모니터링, 풍력 발전 예측), 전파천문학, 싱크로트론을 이용한 엑스선 회절, 사기 방지 등 다양한 분야에서 InfoSphere Streams 활용 사례가 급속히 늘고 있습니다.

## 구조 개요

InfoSphere Streams 아키텍처는 컴퓨터 시스템의 조직과 기능에 심대한 변화를 의미합니다. CEP(Complex Event Processing) 시스템과 일부 유사한 측면이 있으나, 더 높은 데이터 속도와 더 많은 데이터 유형을 지원합니다. 또한 일정 수립, 부하 조정, 고가용성 등의 동적 적응성 및 확장성 요구를 해결하는 인프라 지원도 제공합니다.

InfoSphere Streams에서 연속적인 애플리케이션은 하나 이상의 데이터 스트림을 기반으로 상호 연결되어 운영되는 개별 연산자로 구성됩니다. 일반적으로 데이터 스트림은 시스템 외부에서 유입되거나, 애플리케이션의 일부로 내부적으로 생성될 수 있습니다. 아래의 흐름도는 다수의 소스와 다양한 스트리밍 데이터를 필터링, 분류 및 변환하고, 상관 관계를 분석하고, 상호 융합하면서 소득 관련 뉴스 분석에 따라 적절히 수정되는 동적 소득 계산과 임박한 허리케인 피해의 영향 등의 실시간 위험 평가를 통해 주식 거래 관련 결정을 수행하는 과정을 보여줍니다.



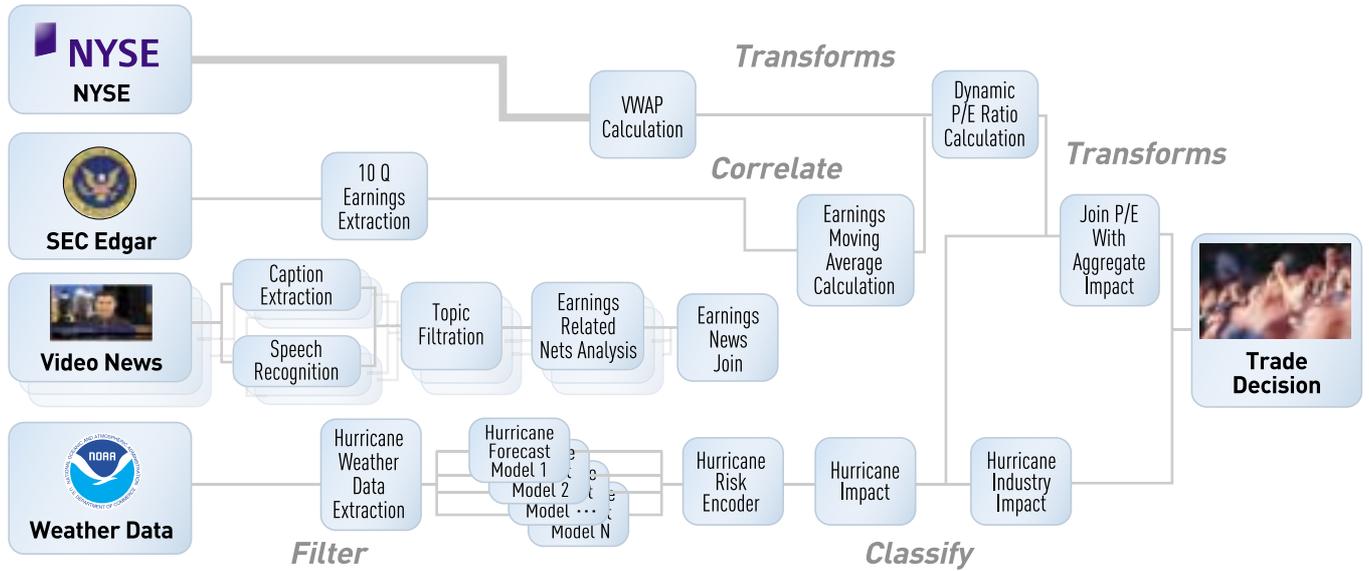


그림 2: 주식 거래의 예

이 개요의 논지를 파악하기 위해 그림 2의 세부적인 사항들을 모두 이해할 필요는 없습니다. InfoSphere Streams 외부의 스트리밍 데이터 소스가 InfoSphere Streams 시스템의 핵심부로 들어와 여러 애플리케이션에 의해 다양한 방식으로 분석되고, 시스템의 여러 부분을 거쳐 결과를 산출하는 과정을 제시하는 것이 이 그림의 목적입니다. 이러한 산출 결과는 상황판에 표시되거나, 사업적 조치에 반영되거나, 추후 이력 분석을 위해 기업 데이터베이스 또는 웨어하우스에 저장되는 등 다양한 방식으로 활용됩니다.

어느 대학의 교수가 큰 유리 단지에 굵은 돌을 채운 다음 단지가 가득 찼는지 학생들에게 물었습니다. 학생 대부분은 가득 찼다고 대답했습니다. 교수는 계속해서 작은 조약돌을 넣었습니다. 단지를 흔들자 상당한 수의 조약돌을 더 넣을 수 있었습니다. 교수는 다시 단지가 가득 찼는지 물었습니다. 앞선 학습 효과로 인해 대부분의 학생들은 아니라고 답했습니다. 교수는 다시 단지를 흔들면서 작은 틈새를 모두 채워 가면서 단지의 맨 윗부분까지 모래를 부었습니다. 교수는 다시 물었습니다. “단지가 가득 찼습니까?” 여러 학생들이 굵은 돌이나 조약돌, 모래 등을 더는 넣을 수 없을 것으로 생각하여 단지가 가득 찼다고 대답했습니다. 그러자 교수는 물병을 꺼내 단지에 물을 부었습니다. 교수는 이 단지는 인생과 비슷하며, 사람이 인생을 살면서 바위나 큰 돌처럼 중요한 것부터 먼저 채워 넣지 않으면 나중에 더 넣을 수 있는 기회를 결코 찾지 못하게 될 것이라고 설명했습니다.

그런데 이 이야기에서는 다른 원리도 찾을 수 있습니다. 작은 구성요소를 이용할수록 낭비되는 공간이 줄어듭니다. 물 분자는 모래가 채워진 단지 내의 눈에 보이지 않는 공간까지 쉽게 채울 수 있습니다. Streams는 이 원리를 이용하여 성능과 지연 시간을 최적화 합니다. 스트리밍 애플리케이션에서는 매우 미세한 연산자가 사용되는데 이들이 융합되어 처리 요소(Processing Elements)를 이루고, Streams에 배치되어 실행됩니다. 그리고 고급 컴파일러가 높은 수준의 선언적 SPL(Streams Processing Language)을 즉시 실행 가능한 기계어로 변환합니다. 또한 이 컴파일러는 stateless 및 stateful 연산자를 감지하여 멀티코어 컴퓨터에서 다수의 스레드를 사용할 수 있도록 합니다. 이러한 앞선 기능으로 개발자들은 다수 코어 프로그래밍 문제를 신속히 해결할 수 있을 뿐 아니라, 탁월한 성능과 지연 시간이 매우 적은 프로세싱을 구현할 수 있습니다.



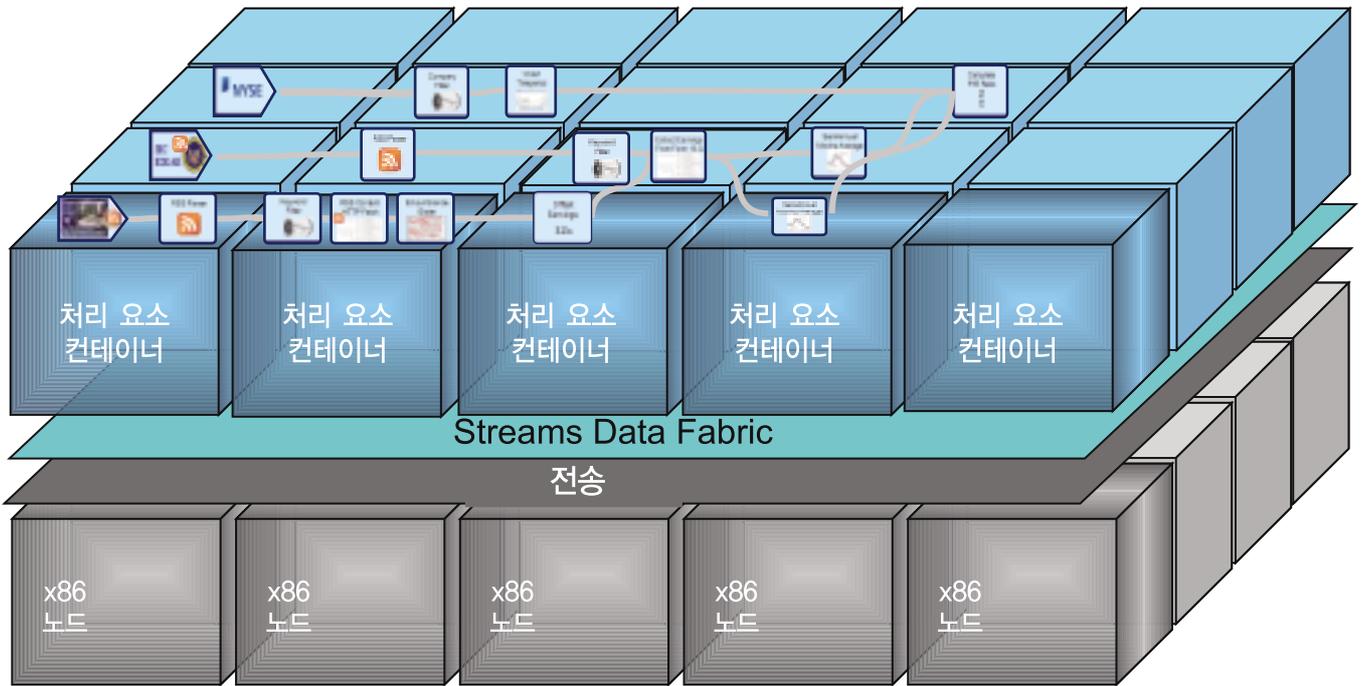


그림 3: 시스템 개요도

그림 3은 매우 큰 규모의 x86 아키텍처 컴퓨터 클러스터를 지원하는 InfoSphere Streams 인프라를 보여줍니다. 그림에 제시된 바와 같이, 매우 다양한 데이터 유형과 양식을 반영하는 입력 데이터 스트림의 데이터가 시스템에 유입됩니다. 관리 서비스 단위에서는 Streams Data Fabric을 이용해서 물리적 전송 계층으로 통신을 수행합니다. 물리적 전송은 10G Ethernet 등의 고속 이더넷이나 더욱 빠른 InfiniBand 전송일 수 있습니다. Streams 관리 서비스에서는 각 연산자, 처리 요소, 작업 및 노드의 성능을 지속적으로 실시간 모니터링 함으로써 작업의 배포를 최적화합니다. 이 정보는 컴파일러가 저장된 성능 정보를 이용해 연산자 융합을 최적화하는 경우 애플리케이션 개발 단계에 있어 특히 중요합니다.

또한 그림 3은 Streams에 다수의 작업을 배포할 수 있음을 보여줍니다. 이 새로운 작업들은 동적으로 추가될 수 있으며, 작업들 사이에서 데이터 스트림을 내보내거나 가져올 수 있습니다. 기존 애플리케이션에 새로운 작업을 추가하여 새로운 방식으로 데이터를 분석할 수 있기 때문에 이러한 융통성은 기업의 기민성을 크게 높여줍니다. 또한 시스템을 재시작할 필요 없이 Streams 런타임에 새로운 입력 스트림과 출력 스트림, 노드를 동적으로 추가 또는 제거하여 용량 증가에 대응할 수 있습니다.



InfoSphere Streams로 최종 사용자들은 다음과 같이 다양한 방식으로 스트리밍 데이터를 사용할 수 있습니다.

- SPL(Stream Processing Language)은 Streams 런타임 프레임워크와 연동하여 스트리밍 애플리케이션을 지원하는 언어를 제공합니다. 사용자들은 스트림별 작업을 상세한 수준까지 파악하지 않고도 애플리케이션을 만들 수 있습니다. SPL은 다양한 내장형 연산자와, InfoSphere Streams 외부에서 스트림을 가져오고 결과를 시스템 외부로 내보내는 기능, 사용자가 정의한 연산자로 시스템을 확장하는 기능을 제공합니다.
- 사용자들은 Eclipse 기반 통합 개발 환경(IDE)인 Streams Studio를 이용해 애플리케이션을 개발할 수 있습니다. 이 사용자들은 스트림을 통해 상호 연결할 수 있는 낮은 레벨의 애플리케이션 구성요소를 프로그래밍하고, 이러한 연결의 성격을 지정할 수 있습니다. 각 구성요소는 "유형화"되므로 나중에 다른 구성요소가 특정 스트림을 재활용하거나 새로 만들 수 있습니다. Java 또는 C++에서 개발된 기존 분석 루틴을 이러한 Streams 애플리케이션에 들여올 수 있으므로 개발에 소요되는 시간을 단축하고, 기존 알고리즘을 재활용할 수 있습니다.
- 그리고 사용자들은 Streams 애플리케이션에 기존 데이터 마이닝 스키어링 모델을 배포하여 실시간 통찰을 확보할 수 있으며, 이에 따라 영구적인 또는 저장된 데이터에 이러한 모델을 실행할 필요가 없습니다. 또한 Streams 애플리케이션은 앞서 사이버 보안 사례에서 확인된 바와 같이 새로운 모델의 필요성을 감지하고, 업데이트된 모델의 생성을 촉발하며, 실행 중인 애플리케이션에 새로운 모델을 배포할 수 있습니다.

이러한 모든 기능은 실행 중인 런타임 시스템에 의해 지원됩니다. 새로운 작업이 제출되면 InfoSphere Streams 스케줄러는 새로 제출된 명세와 기존 명세의 요건을 모두 충족하는 최적의 방법을 파악하며, Job Manager는 필요한 변경을 자동으로 실행합니다. 런타임은 컴퓨팅 리소스의 상태와 이용 내역뿐 아니라, 사용자들의 정보 요구와 그 요구에 부합하는 데이터의 가용성까지 지속적으로 모니터링 하면서 그것에 맞춰 적응합니다.

실행 중인 애플리케이션에서 도출된 결과는 InfoSphere Streams 외부에서 실행되는 프로세스(웹 서버 등)를 통해 처리됩니다. 예를 들어, 애플리케이션에서는 TCP 연결을 통해 상시적인 데이터 스트림을 수신하여 맵에 시각화하거나, 관리자에게 변칙적이거나 "흥미로운" 이벤트가 있음을 알릴 수 있습니다.



## Streams 2.0의 새로운 특징

IBM InfoSphere Streams V2.0은 다음과 같이 다양한 기능과 특징을 통해 애플리케이션 개발 속도를 높이고, Streams의 성능과 가용성, 관리 기능을 개선합니다.

### 스트리밍 애플리케이션 개발 작업의 단순화

IBM InfoSphere Streams Studio는 Eclipse 기반 톨로서, SPL(Streams Processing Language) 및 SPL 혼합 모드 애플리케이션을 생성, 수정, 시각화, 테스트, 디버그 및 실행할 수 있도록 합니다. Streams Studio의 통합 개발 환경(IDE)을 구성하는 주된 요소는 다음과 같습니다.

- Streams 개발 환경의 설정 및 관리에 유용한 Streams Explorer 뷰
- SPL 애플리케이션과 툴킷의 구성 및 구축을 위한 SPL Project 및 SPL Application Set Project 지원
- 자동 완성, 코드 템플릿, 코드 폴딩, 연속 빌드, 인라인 오류 보고, 리팩터링(refactoring), 개요 뷰 등의 기능을 지원하는 SPL 편집기
- 분할 화면 SPL 혼합 모드 편집기
- 툴킷 모델, 연산자 모델, 기능 모델 편집기
- 런타임 인스턴스를 관리할 수 있는 Streams Explorer
- 합성 연산자 펼치기/접기를 지원하는 비주얼라이저
- SPL 리소스를 논리적 방식으로 시각화하는 Project Explorer 뷰
- 애플리케이션 토폴로지를 보여주는 그래프 뷰
- 실행 중인 애플리케이션의 메트릭을 조회 및 분석할 수 있는 메트릭 뷰
- 실행 중인 인스턴스에서 로그를 수집해서 검사할 수 있는 로그 조회 지원 기능
- 단독형 및 분산형 애플리케이션의 실행을 위한 실행기
- Streams 애플리케이션의 테스트 및 디버깅을 위한 디버거
- SPL 애플리케이션 및 툴킷의 개발 과정에서 필요한 정보를 제공하는 내장 도움말 시스템

InfoSphere Streams V2.0에는 개발자의 생산성 향상으로 비즈니스 민첩성을 크게 높일 수 있는 매우 다양한 기능 개선이 포함되어 있습니다. 새로이 개선된 SPL(Streams Processing Language)의 주요 개선 사항은 다음과 같습니다.

- 구문론, API 및 기타 구성의 일관성 향상
- 데이터 항목을 계층적으로 표현할 수 있는 유형 시스템(type system)
- 데이터를 계층적으로 표현할 수 있는 중첩형 유형
- 원시 연산자 및 네이티브 함수를 쉽게 작성할 수 있는 확장 메커니즘
- 대규모 개발을 위한 합성 연산자



- 연산자 행동의 즉각적 업데이트 지원
- 연산자 개발자가 사용할 수 있는 공용 windowing 라이브러리
- 목록, 집합, 맵 등의 공용 컨테이너 유형
- 멀티스레드 흐름 생성을 위한 컴포지션 레벨 구성물
- 네임스페이스 및 복수 파일을 통한 모듈성 지원
- 연산자 및 함수를 위한 versioned 툴킷 지원
- 새로운 관계형, 어댑터, 유틸리티 연산자를 갖춘 새로운 표준 툴킷
- compile-time 성능 개선(compile-time 폴딩 및 function 계산은 코드 생성 작업을 최소화하고, 성능까지 개선함)

### 더 빠른 속도로 더 많은 양을 처리하게 해주는 성능 향상

InfoSphere Streams 런타임은 구조화 정보와 비구조화 정보를 모두 빠른 속도로 수집 및 분석하여 비즈니스 통찰을 제공합니다. Streams 프로그램의 일부가 런타임 컴퓨팅 클러스터의 여러 노드에 분산되어 초당 수백만 통의 메시지 처리 용량과 밀리초 이하의 속도를 구현합니다. Streams V2.0은 정보 처리량이 갈수록 늘어나 처리 속도를 높여야 하는 기업들에게 도움이 되기 위해 다음과 같이 지속적으로 성능을 개선하고 있습니다.

- 첨단 멀티스레드 하드웨어를 더욱 효과적으로 활용할 수 있는 새로운 멀티스레딩 지원
- Java 지원 개선을 통한 공유형 Java Virtual Machines의 메모리 활용도 개선
- 인트라노드 및 인터노드 커뮤니케이션 개선을 통한 Streams의 데이터 처리 성능 향상 런타임(연산자 구현, 런타임 퓨전, 전송 통합)

### 시스템 관리 개선을 통한 관리 비용 절감 및 런타임 용량 개선

InfoSphere Streams에서는 명령형 인터페이스는 물론 그래픽 인터페이스를 통해 Streams 런타임을 관리하고, 최적의 애플리케이션 성능과 가용성을 유지할 수 있습니다. 웹 기반 관리 콘솔에서는 클러스터에 속한 여러 노드의 인스턴스를 생성, 시작 및 정지할 수 있으며, 이 인스턴스의 애플리케이션을 취소할 수 있습니다. 또한 Streams 런타임 활용을 최적으로 유지하기 위해 노력하는 관리자들의 작업을 간소화하는 새로운 기능과 향상된 기능이 다수 제공됩니다.

- 새로운 상태 메트릭의 프로그래밍을 통해 애플리케이션 성능과 가용성을 쉽게 개선할 수 있음
- Processing Element 레벨이 아닌, Streams 연산자 레벨까지 내려가는 더욱 세밀한 상태 메트릭 및 연산자 기능
- 재컴파일 없이 애플리케이션을 사용자 정의할 수 있는 제출 시간 값
- 인스턴스에 관계 없이 애플리케이션을 이동할 수 있어 런타임 용량 향상에 효과적
- 호스트 풀의 동적 할당을 통한 런타임 플랫폼 기민성 향상
- 호스트 태깅을 통한 새로운 ex-location 및 isolation 제약으로 작업(job) 배포 간소화
- 가져오기 구독의 프로그래밍 처리를 위한 런타임 API
- 일반적인 창 조작을 위한 런타임 및 코드 생성 API
- 새로운 연산자 수준 정보를 제공하는 Streams Administrative 콘솔



## 고가용성 기능

InfoSphere Streams는 고가용성과 중복성을 강화하는 몇 가지 기능을 갖고 있습니다. 관리자는 클러스터에 프로세서 노드를 추가하거나 제거할 수 있습니다. 클러스터에 노드를 원활하게 넣고 뺄 수 있으므로 InfoSphere Streams 애플리케이션을 종료하지 않고, 운영체제 또는 데이터 스트림에 필요한 정비 작업을 수행할 수 있습니다. 따라서 환경의 전반적 가용성 향상에 도움이 됩니다. 또한 (시스템을 제어하는) 관리 노드에 문제가 발생하더라도 애플리케이션 노드는 계속 InfoSphere Streams 애플리케이션을 실행할 수 있으며, 관리 노드는 나중에 관리자가 재시작할 수 있습니다. 새로운 프로그래밍을 통한 스트리밍 데이터 조작 기능은 예비 애플리케이션 구성요소를 이용하여 쉽게 고가용성 애플리케이션을 만들 수 있게 해줍니다. 호스트 풀의 동적 할당과 런타임 클러스터의 노드 위치 파악 및 격리 제약 요소를 이용하면 예비용 애플리케이션 구성요소를 격리하여 런타임에 노드를 분리할 수 있습니다. 이와 같은 기능들은 InfoSphere Streams와 관련된 고가용성 향상을 위한 것입니다.

## 요약

InfoSphere Streams는 발표된 지 얼마 지나지 않아 150곳이 넘는 고객사에 설치되어 폭넓은 성공을 입증했습니다. InfoSphere Streams는 빠른 속도로 미션 크리티컬 데이터를 분석할 수 있는 인프라를 제공합니다. IBM은 앞으로도 이 분야에 대한 대폭적인 투자를 통해 이 기술의 효과와 확장성, 보안성, 적실성을 향상시키고, 기존 정보 인프라와의 상호운용성을 강화할 것입니다.

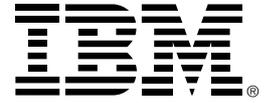
InfoSphere Streams 또는 InfoSphere Streams에 관한 보다 자세한 내용은 IBM 지사 또는 공인 IBM Business Partner에 문의하십시오.

## 추가 정보

IBM InfoSphere Streams 및 그 구축 관련 제품에 관한 자세한 내용은 다음 페이지를 참조하십시오.

<http://www.ibm.com/software/data/infosphere/streams/>





© Copyright IBM Corporation 2011

한국IBM Global Business Services  
(135-270) 서울시 강남구 도곡동 467-12  
군인공제회관빌딩

TEL: (02)3781-7800  
[www.ibm.com/kr](http://www.ibm.com/kr)

2011년 9월

Printed in Korea  
All Rights Reserved

IBM, IBM 로고, ibm은 미국 및/또는 다른 국가에서 IBM Corporation의 상표 또는 등록 상표입니다. 상기 및 기타 IBM 상표로 등록된 용어가 본 문서에 처음 나올 때 상표 기호(® 또는 ™)와 함께 표시되었을 경우, 이러한 기호는 본 문서가 출판된 시점에 IBM이 소유한 미국 등록 상표이거나 관습법에 의해 인정되는 상표임을 나타냅니다. 해당 상표는 미국 외의 다른 국가에서도 등록 상표이거나 관습법적인 상표일 수 있습니다. IBM의 최신 상표 목록은 [ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml) 웹 페이지의 "저작권 및 상표 정보" 부분에서 확인할 수 있습니다.

기타 다른 회사, 제품 및 서비스 이름은 다른 회사의 상표 또는 서비스 표시일 수 있습니다.

이 문서에서 IBM 제품과 서비스를 참조한 경우에도 IBM이 비즈니스를 수행하고 있는 모든 국가에서 해당 제품과 서비스를 제공함을 의미하는 것은 아닙니다.

